AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [0009] with the following amended paragraph:

[0009] Some solutions to this problem, particularly in the case of simple object access protocol messages, involve embedding the security information into each message before securing the entire message. While this keeps security information from being detached (or separately corrupted), another problem arises in that each intermediate computer system may need to un-secure (decrypt) the entire message to find out the message destination. Once the intermediate computer system determines where the message is destined, the intermediate computer system will then re-secure (re-encrypt) the entire message and send the message to either a next intermediate computer system (that will do the same decryption/~~decryption~~re-encrypt), or to the receiving computer system. Of course, this adds a layer of uncertainty since the sending computer system may not trust the intermediate computer system to maintain the appropriate security settings upon un-securing/re-securing (decrypting or re-encrypting) the message.

Please replace paragraph [0010] with the following amended paragraph:

[0010] Other problems with embedding security settings in a message ~~relate~~ related to the customization of the security settings. Generally, simple object access protocol messages are secured with information that identifies the sender, and requires identification of the receiver before the receiver can un-secure (decrypt) the message. Consequently, these types of security settings cannot be configured to limit file access to the message, for example, between only certain time frames during a day. Furthermore, simply packaging the message creation software with more security setting fields does not necessarily increase message security. This is due in part since default fields can be identified by another computer system simply by utilizing the application program used to create the message.